

# ***MEMS and Matlab Metrology***

MetroloGT for Matlab – Joe Ceremuga

MetroloGT++ – Austin Chen

2-D Edge Inspection with Matlab – Meghan Shilling

# ***MetroloGT Toolbox for Matlab***

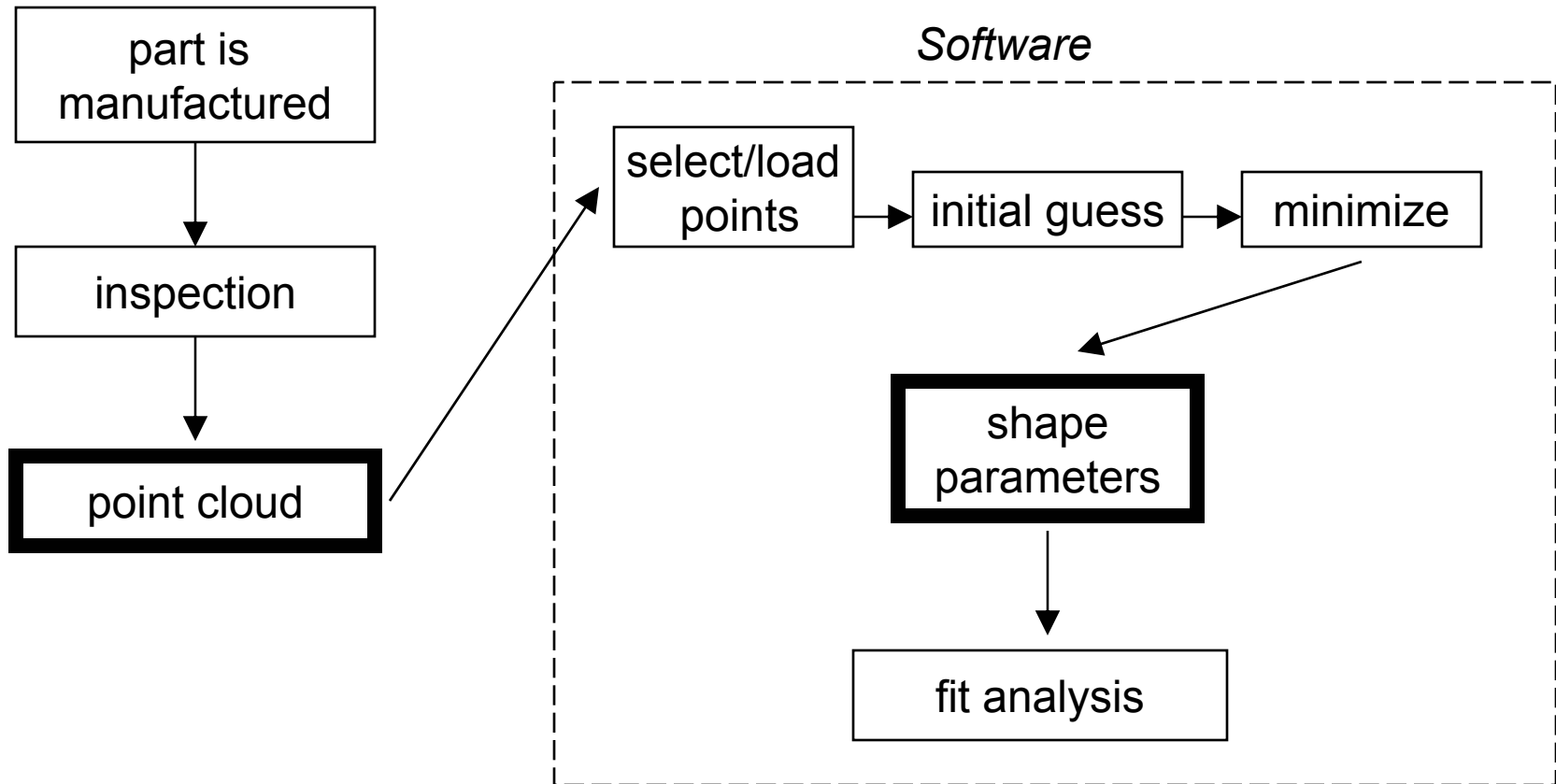
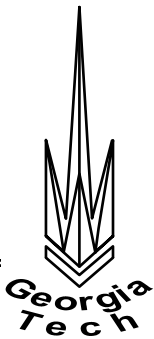


# ***Purpose***

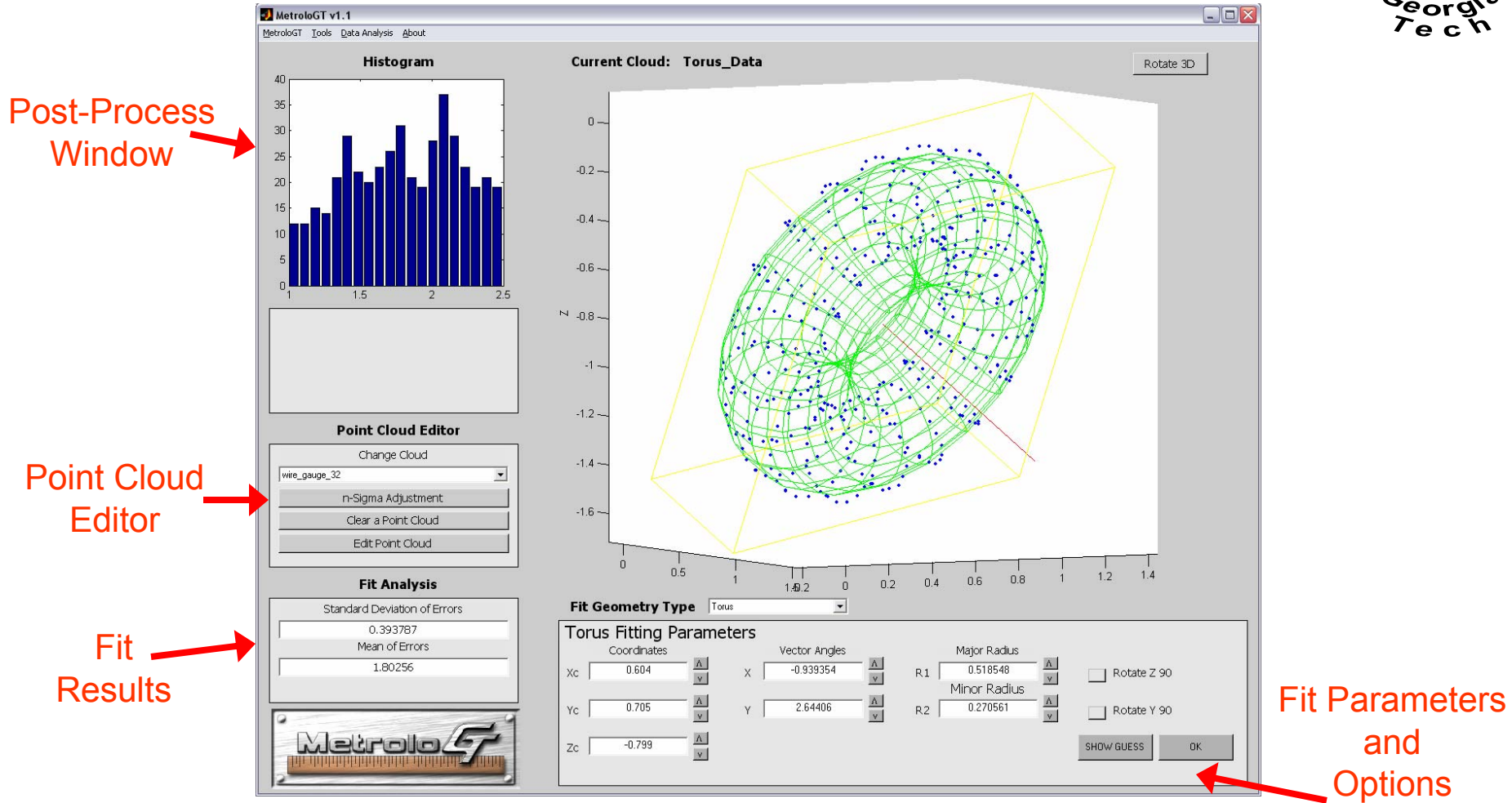
---

- ❖ Input 2-D and 3-D coordinate point clouds
- ❖ Allow the user to manipulate the point cloud
- ❖ Fit primitive shapes to the data
- ❖ Analyze fit and then refine the data for further fitting

# Procedure

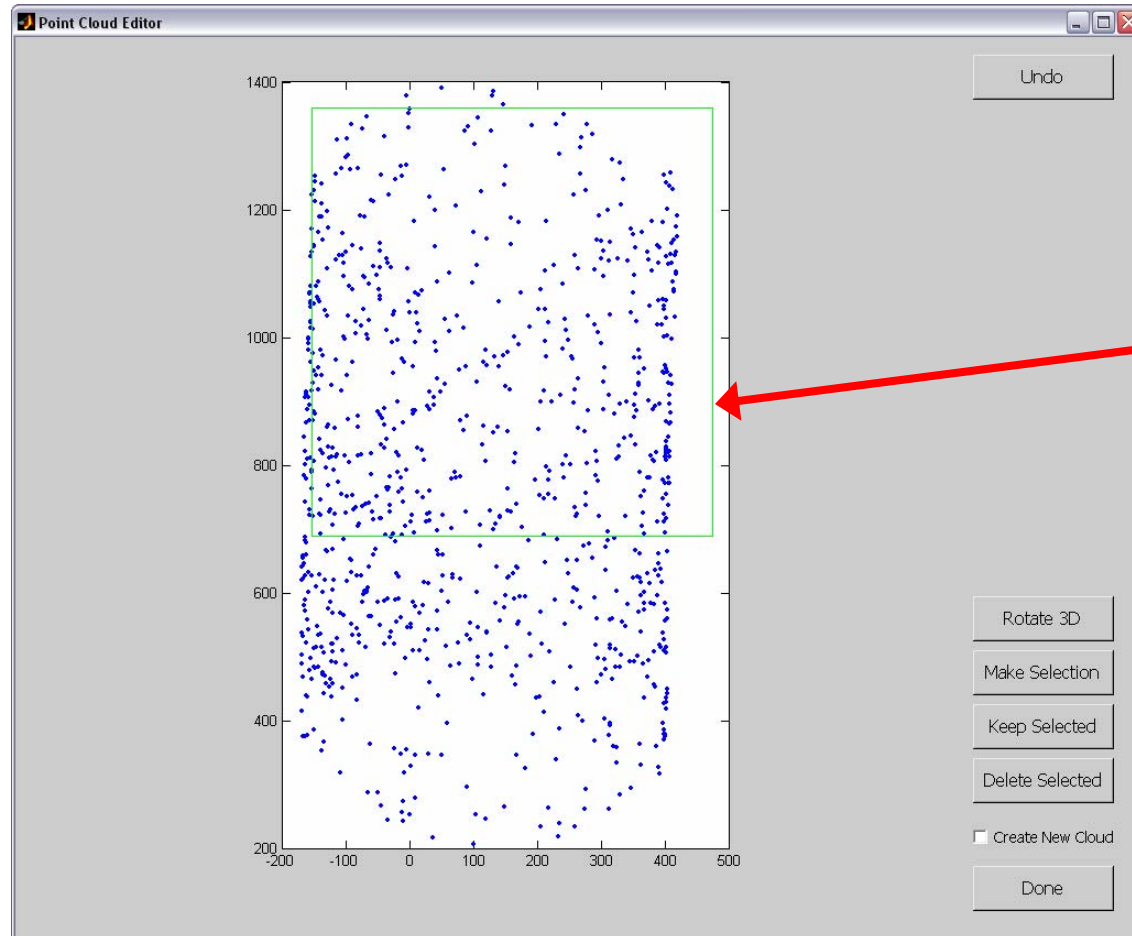


# Software Layout



# Features

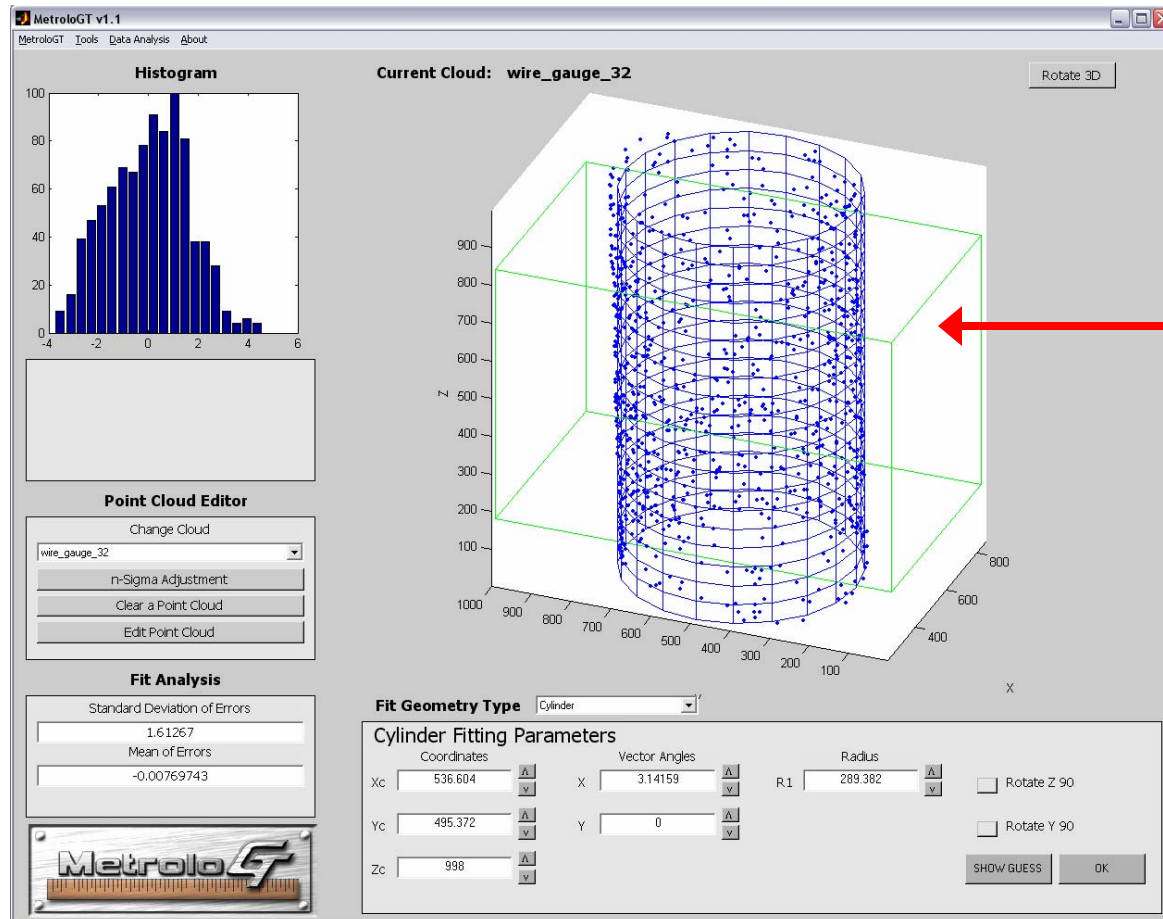
## ❖ Point Cloud Editor



Selection  
Box

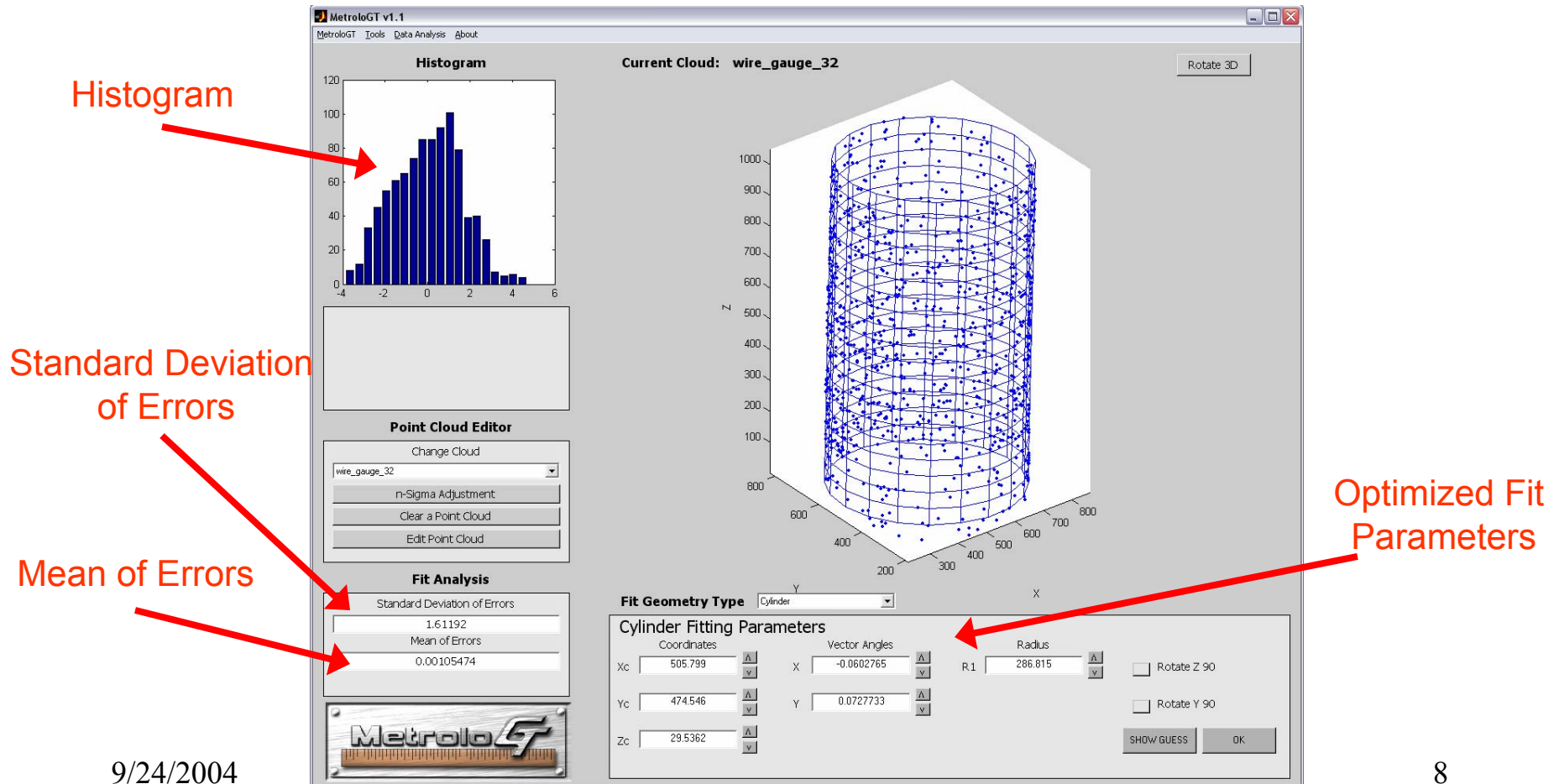
# Features

## ❖ Initial Guessing (Bounding box)



# Features

## ❖ Fit Analysis





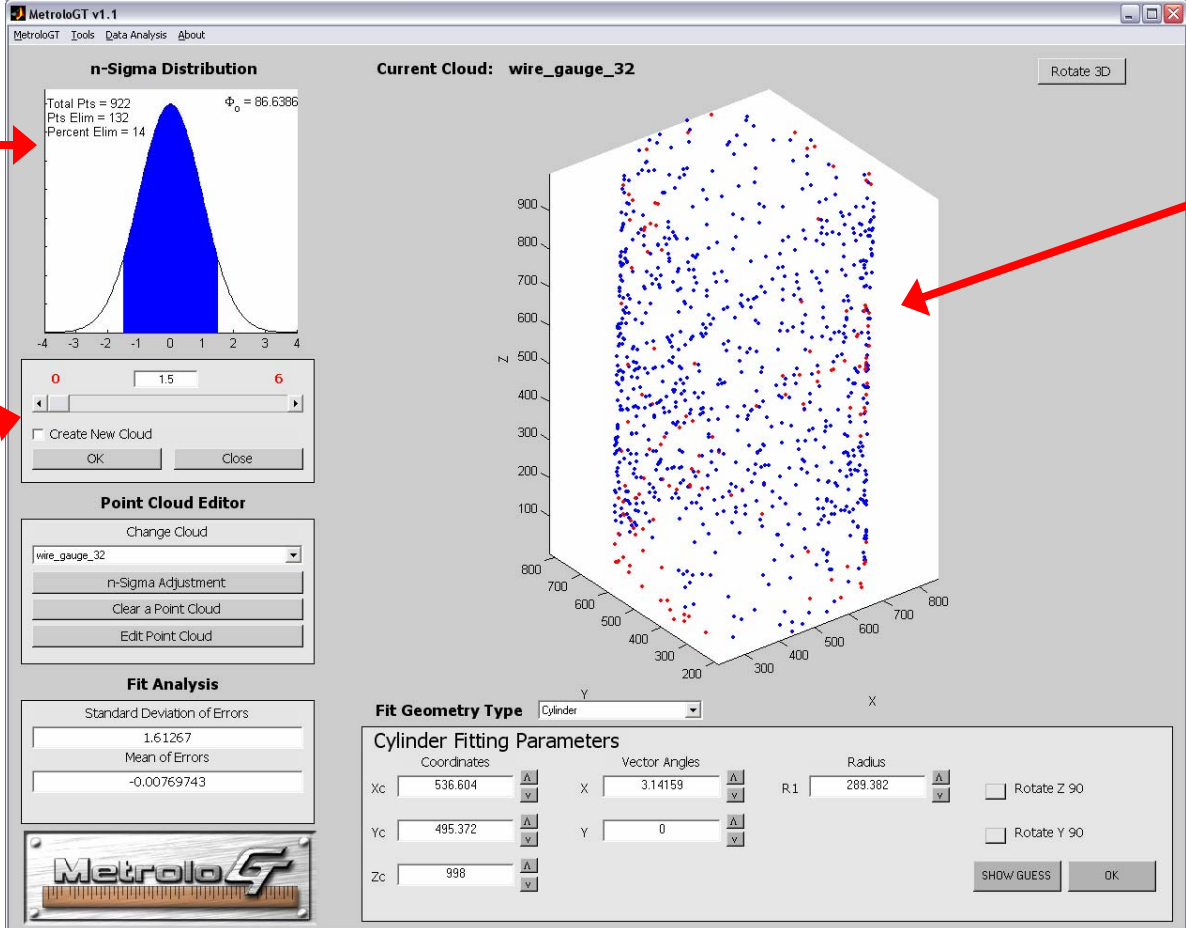
# Features

## ❖ N-Sigma elimination

**Point Elimination Info** →

**N-Sigma Adjustment Control** →

**Points Eliminated (red)** →



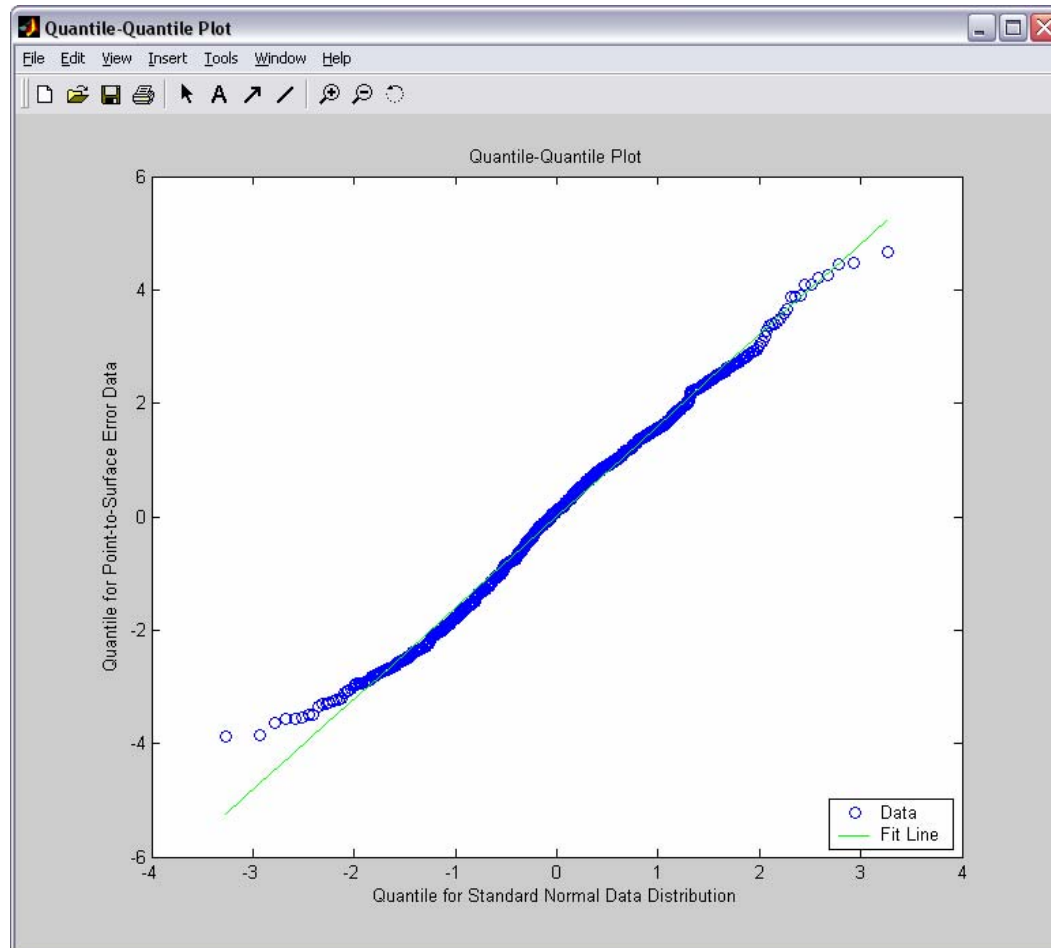
The screenshot displays the MetroloGT v1.1 software interface. The main window shows a 3D point cloud of a wire gauge, with points colored blue (retained) and red (eliminated). The left sidebar contains several panels:

- n-Sigma Distribution:** A histogram showing the distribution of points. It includes statistics: Total Pts = 922, Pts Elim = 132, Percent Elim = 14, and  $\Phi_0 = 86.6386$ .
- N-Sigma Adjustment Control:** A slider set to 1.5, with a range from 0 to 6. It includes a 'Create New Cloud' checkbox and 'OK' and 'Close' buttons.
- Point Cloud Editor:** A section with a 'Change Cloud' dropdown menu (currently set to 'wire\_gauge\_32') and buttons for 'n-Sigma Adjustment', 'Clear a Point Cloud', and 'Edit Point Cloud'.
- Fit Analysis:** A section showing 'Standard Deviation of Errors' (1.61267) and 'Mean of Errors' (-0.00769743).

The main 3D view shows the 'Current Cloud: wire\_gauge\_32'. The axes are labeled X, Y, and Z. A red arrow points to the red points in the cloud, indicating they have been eliminated. The bottom right panel shows 'Fit Geometry Type' set to 'Cylinder' and 'Cylinder Fitting Parameters' including coordinates (Xc, Yc, Zc), vector angles (X, Y), and radius (R1).

# Features

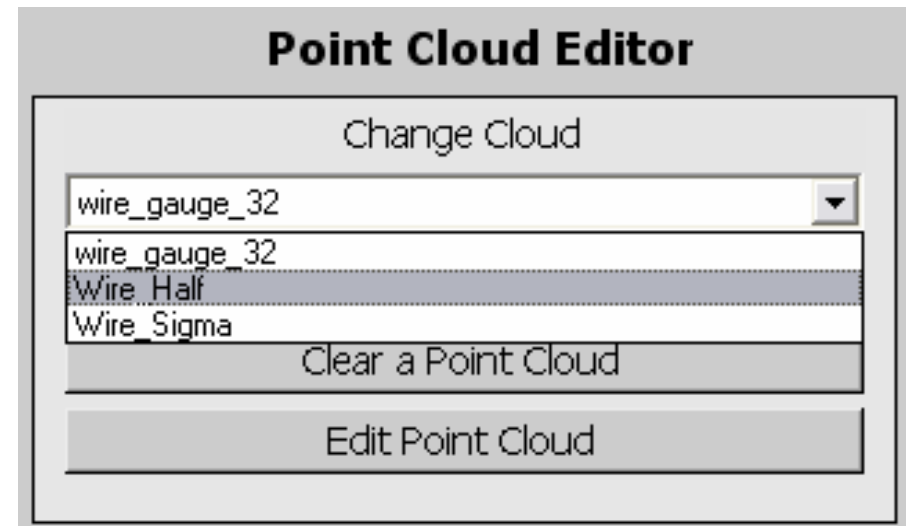
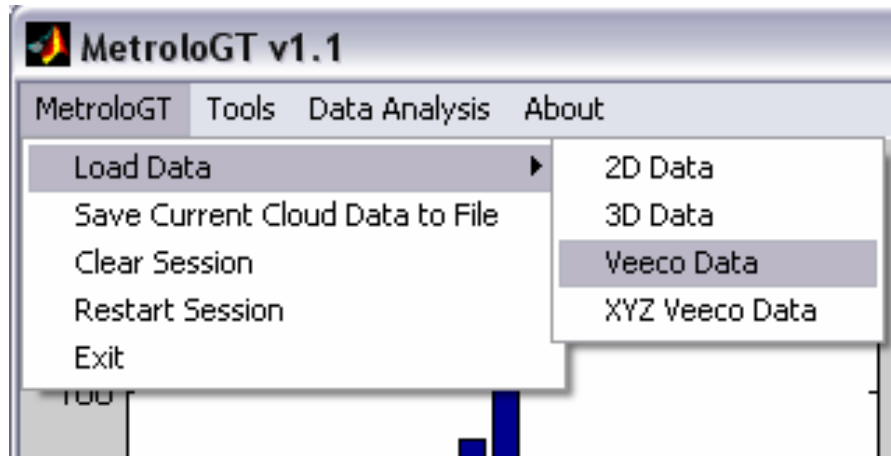
## ❖ Quantile-Quantile (QQ) Plotting



# Features

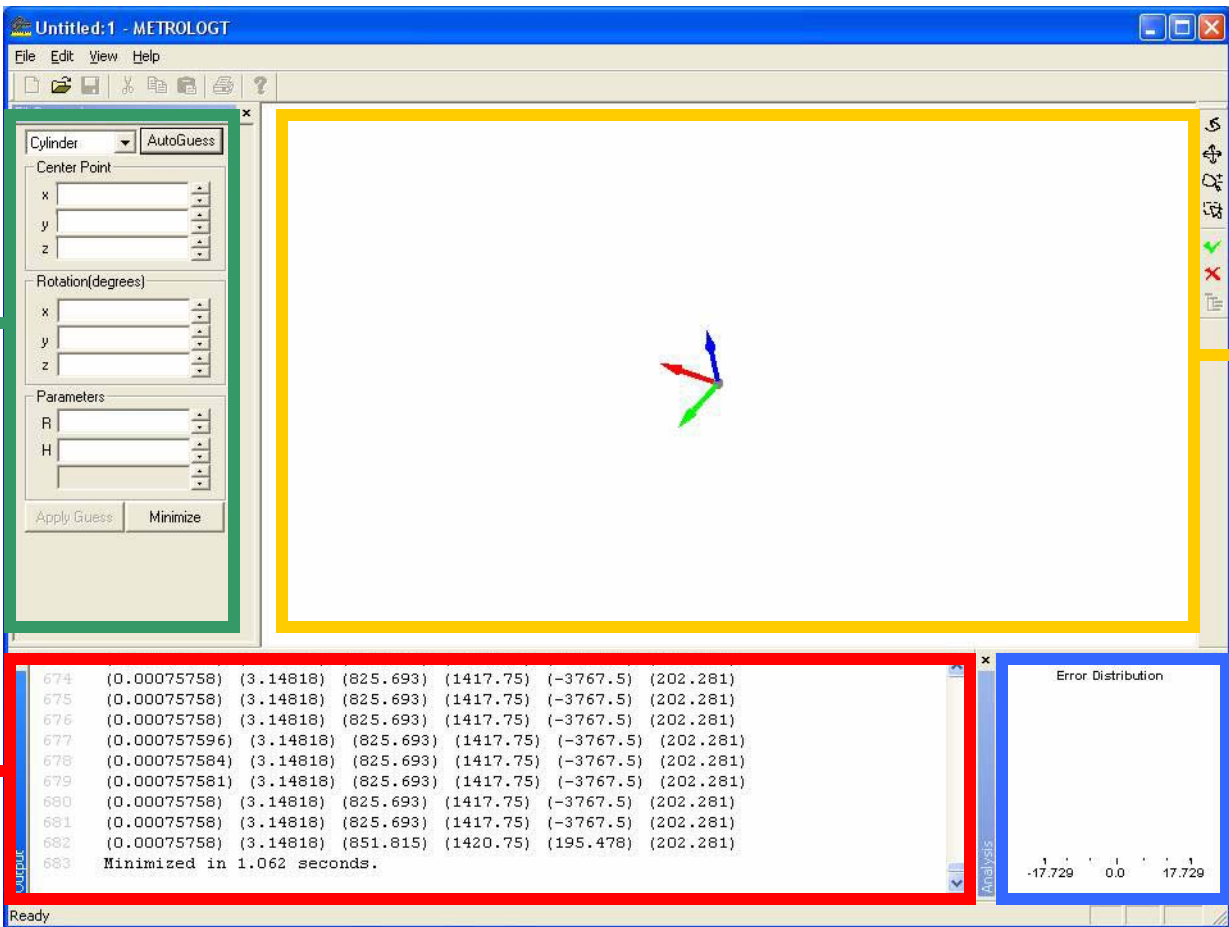
## ❖ Cloud options

- Multiple clouds per session
  - ◆ Load (2-D, 3-D, Veeco <sup>TM</sup> format)
  - ◆ Create (Editor and N-Sigma)
  - ◆ Erase
  - ◆ Save



# ***MetroloGT++***

# MetroloGT++



The screenshot shows the MetroloGT++ software interface. The main window is titled "Untitled:1 - METROLOGT". It features a menu bar (File, Edit, View, Help) and a toolbar. The interface is divided into several sections:

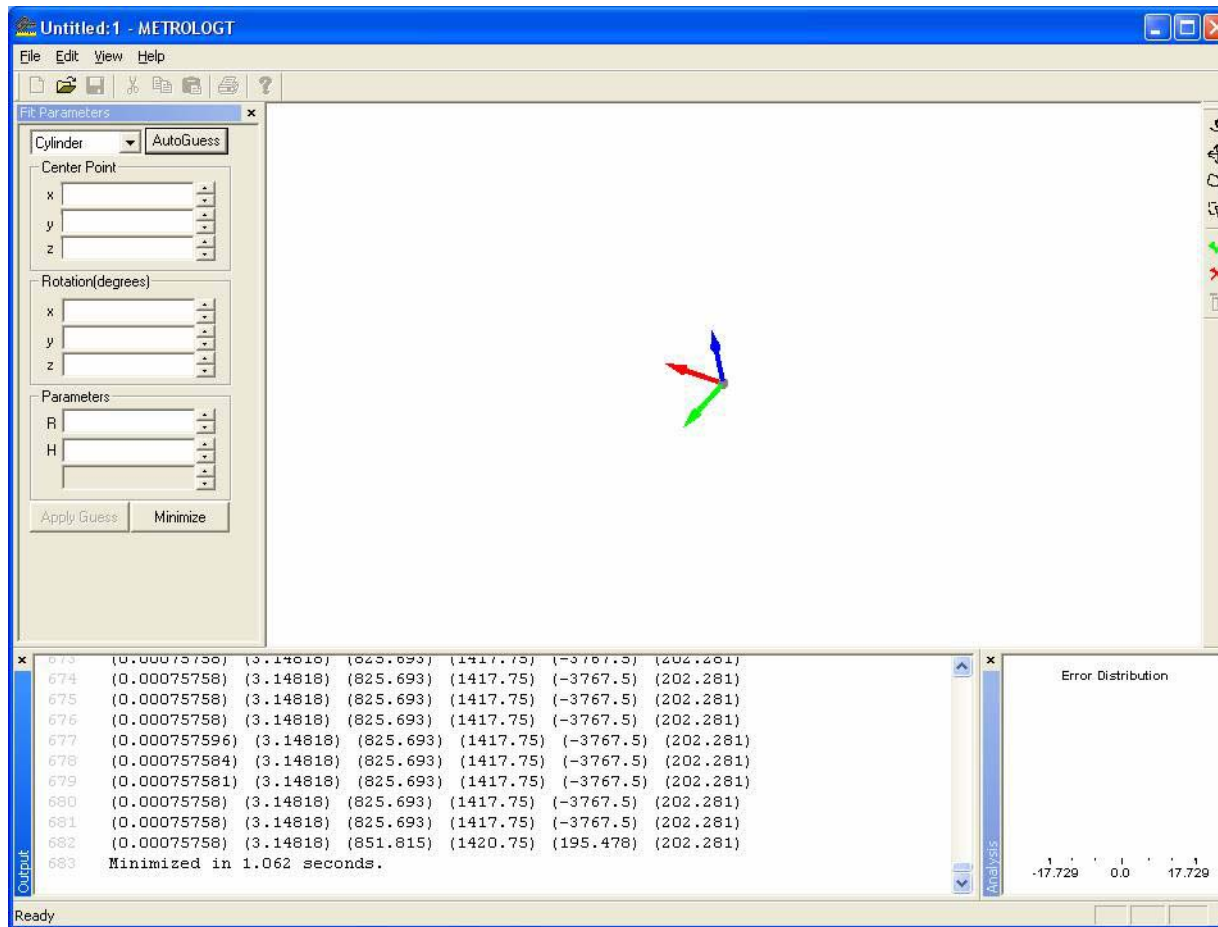
- shape parameters:** A panel on the left containing input fields for "Cylinder" (with an "AutoGuess" button), "Center Point" (x, y, z), "Rotation(degrees)" (x, y, z), and "Parameters" (R, H). It also includes "Apply Guess" and "Minimize" buttons.
- visualization window:** A large central area with a yellow border, displaying a 3D coordinate system with red, green, and blue axes.
- output window:** A panel at the bottom left with a red border, displaying a list of data points and a status message.
- histogram:** A panel at the bottom right with a blue border, titled "Error Distribution", showing a normal distribution curve.

Annotations with colored lines point to these sections: a green line to "shape parameters", a yellow line to "visualization window", a red line to "output window", and a blue line to "histogram".

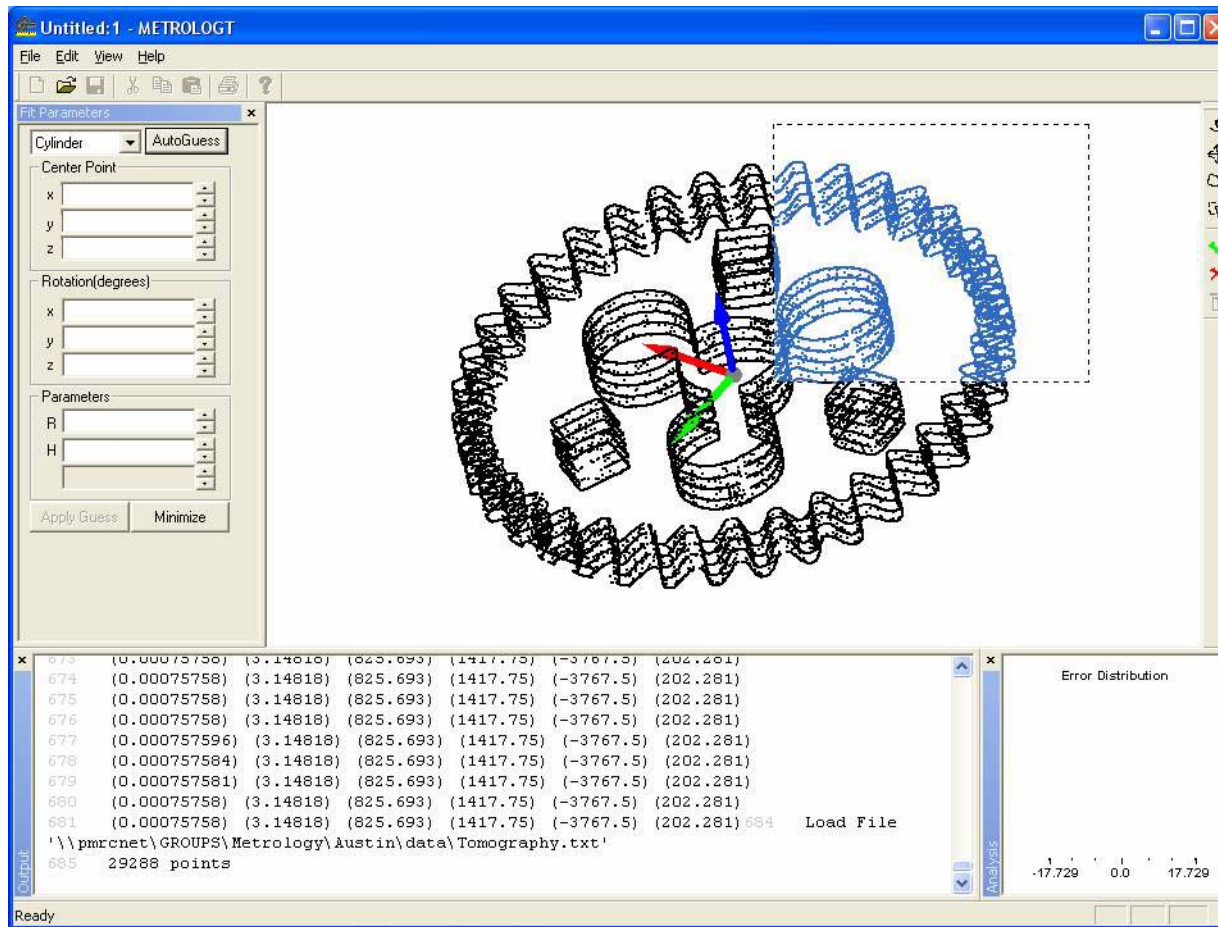
**Output Window Data:**

674	(0.00075758)	(3.14818)	(825.693)	(1417.75)	(-3767.5)	(202.281)
675	(0.00075758)	(3.14818)	(825.693)	(1417.75)	(-3767.5)	(202.281)
676	(0.00075758)	(3.14818)	(825.693)	(1417.75)	(-3767.5)	(202.281)
677	(0.000757596)	(3.14818)	(825.693)	(1417.75)	(-3767.5)	(202.281)
678	(0.000757584)	(3.14818)	(825.693)	(1417.75)	(-3767.5)	(202.281)
679	(0.000757581)	(3.14818)	(825.693)	(1417.75)	(-3767.5)	(202.281)
680	(0.00075758)	(3.14818)	(825.693)	(1417.75)	(-3767.5)	(202.281)
681	(0.00075758)	(3.14818)	(825.693)	(1417.75)	(-3767.5)	(202.281)
682	(0.00075758)	(3.14818)	(851.815)	(1420.75)	(195.478)	(202.281)
683	Minimized in 1.062 seconds.					

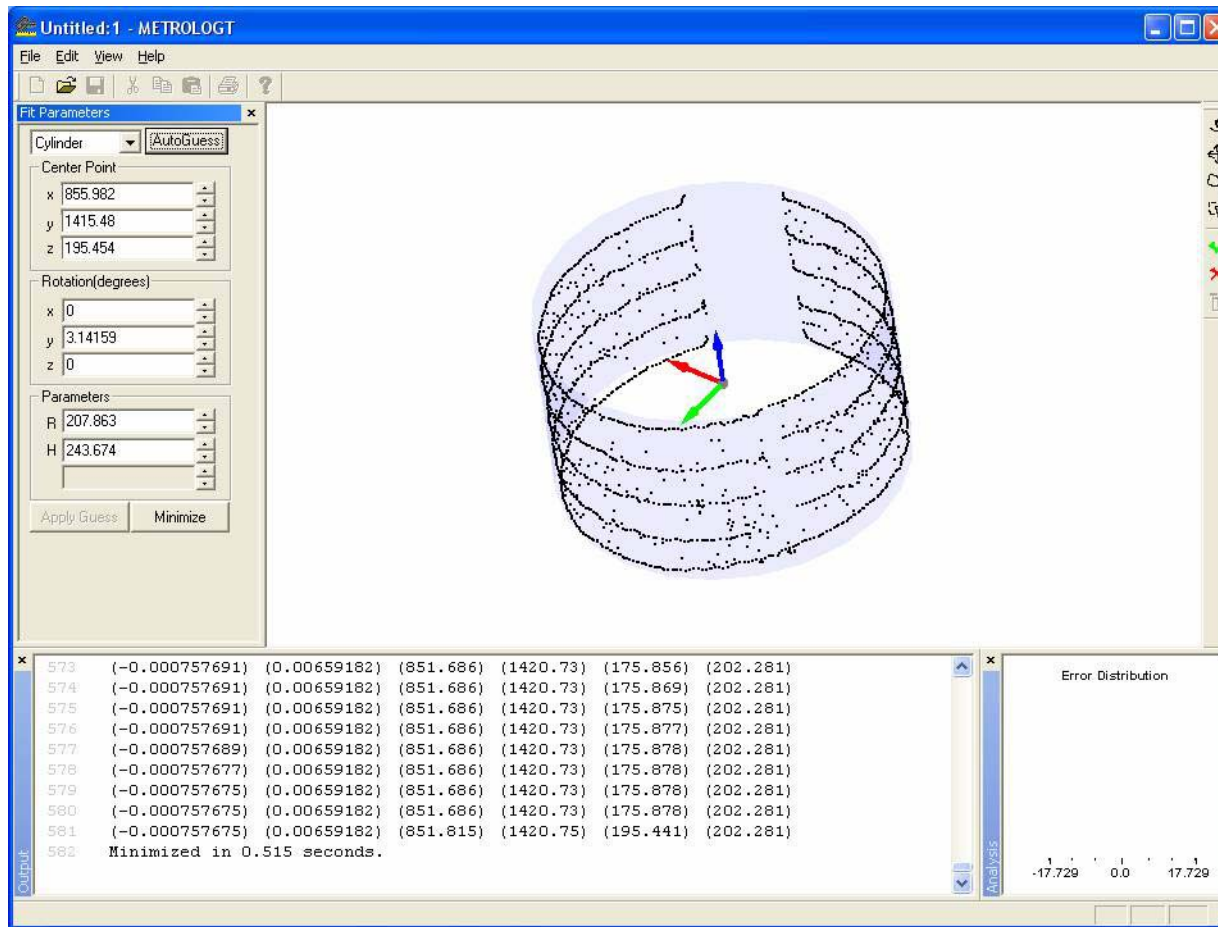
# 1) start



## 2) load and select points

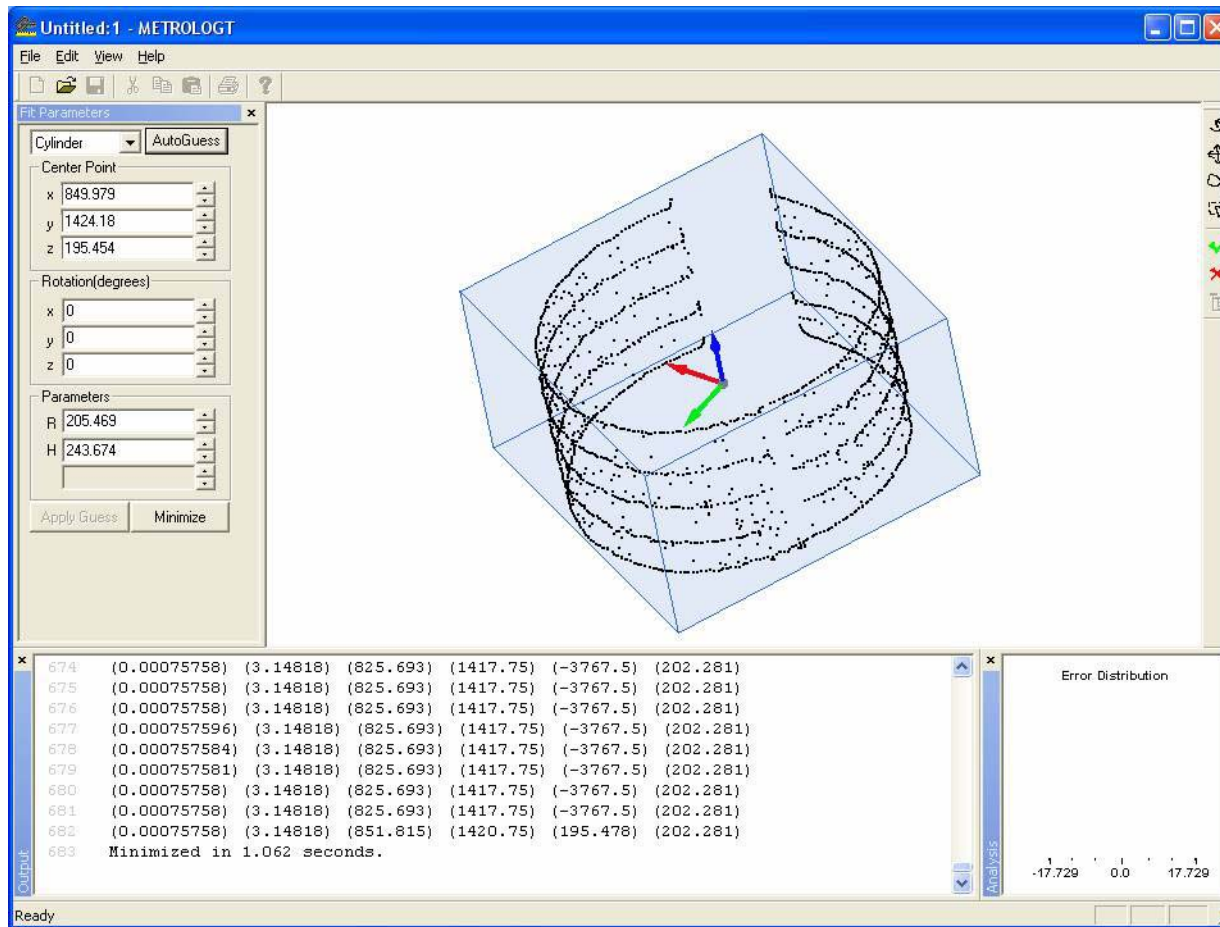


# 3a) *initial guess*

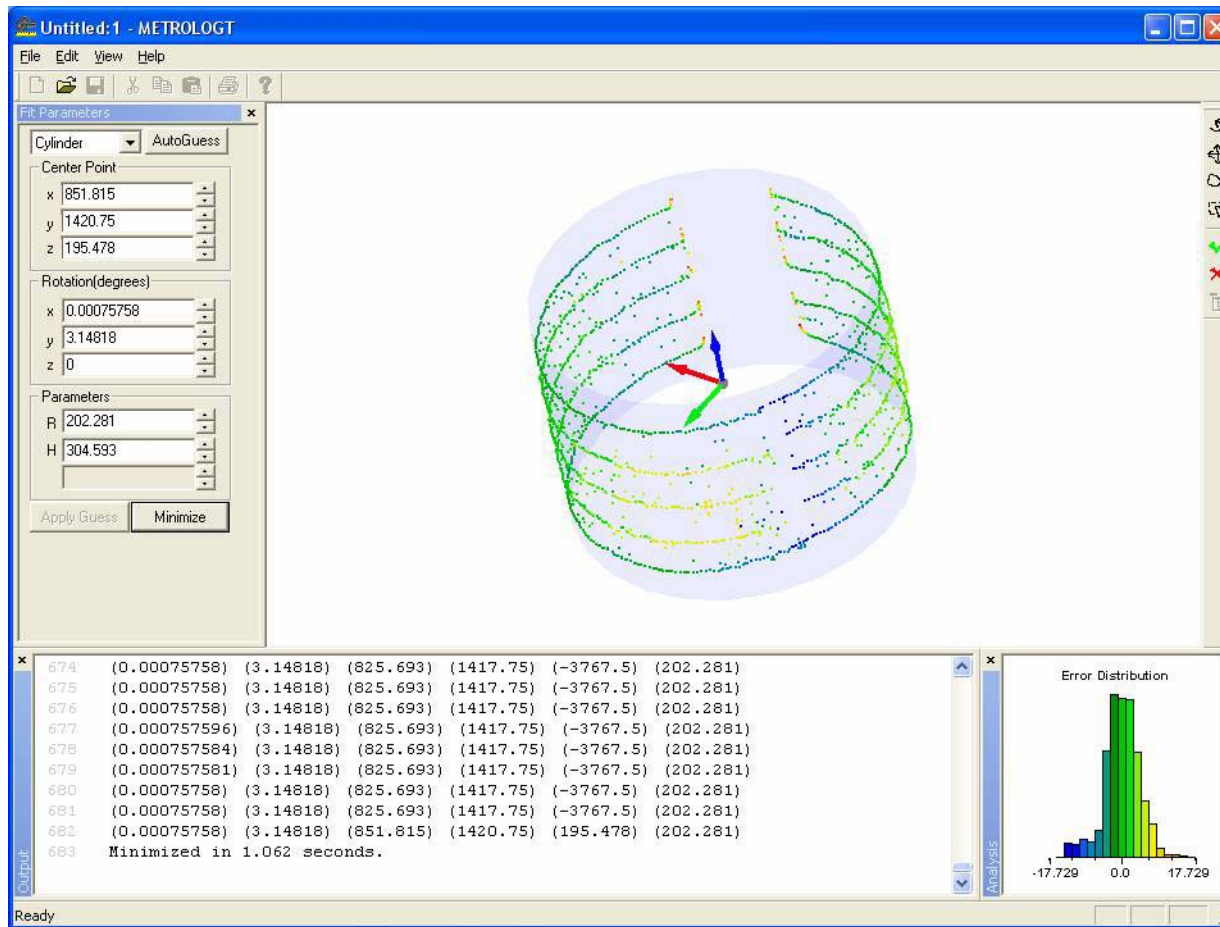




# 3b) *minimum bounding box (hidden)*



# 4) *fit primitive*

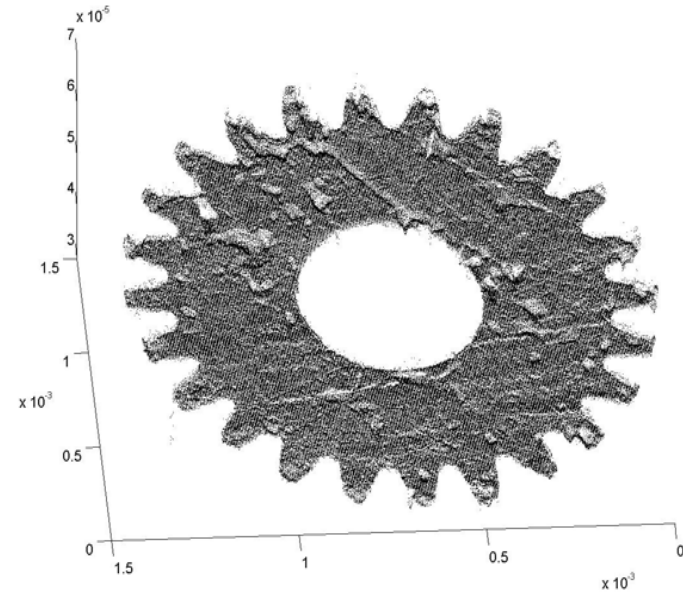


# ***2-D Edge Detection***

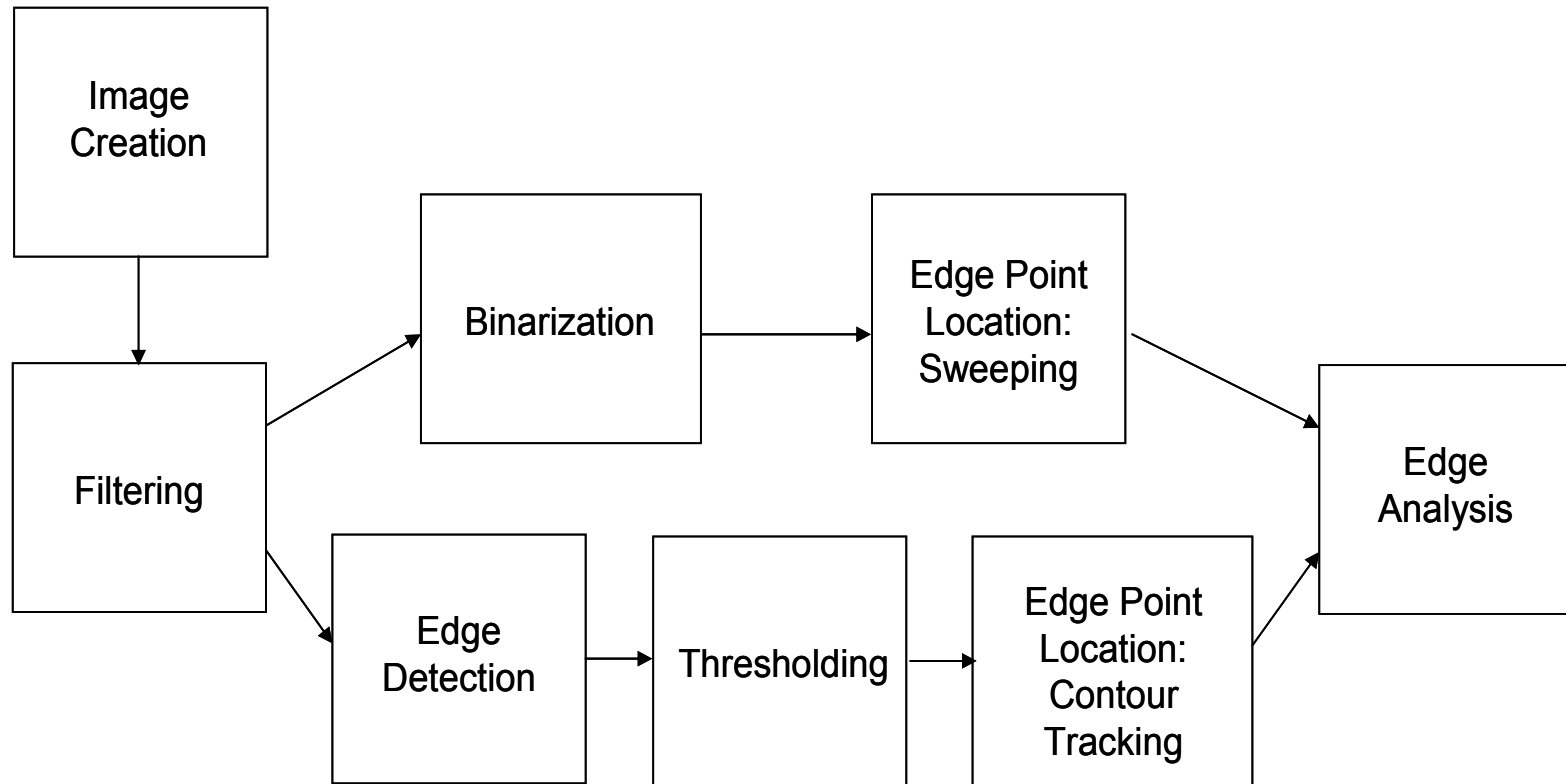
# *Two Dimensional Analysis*

---

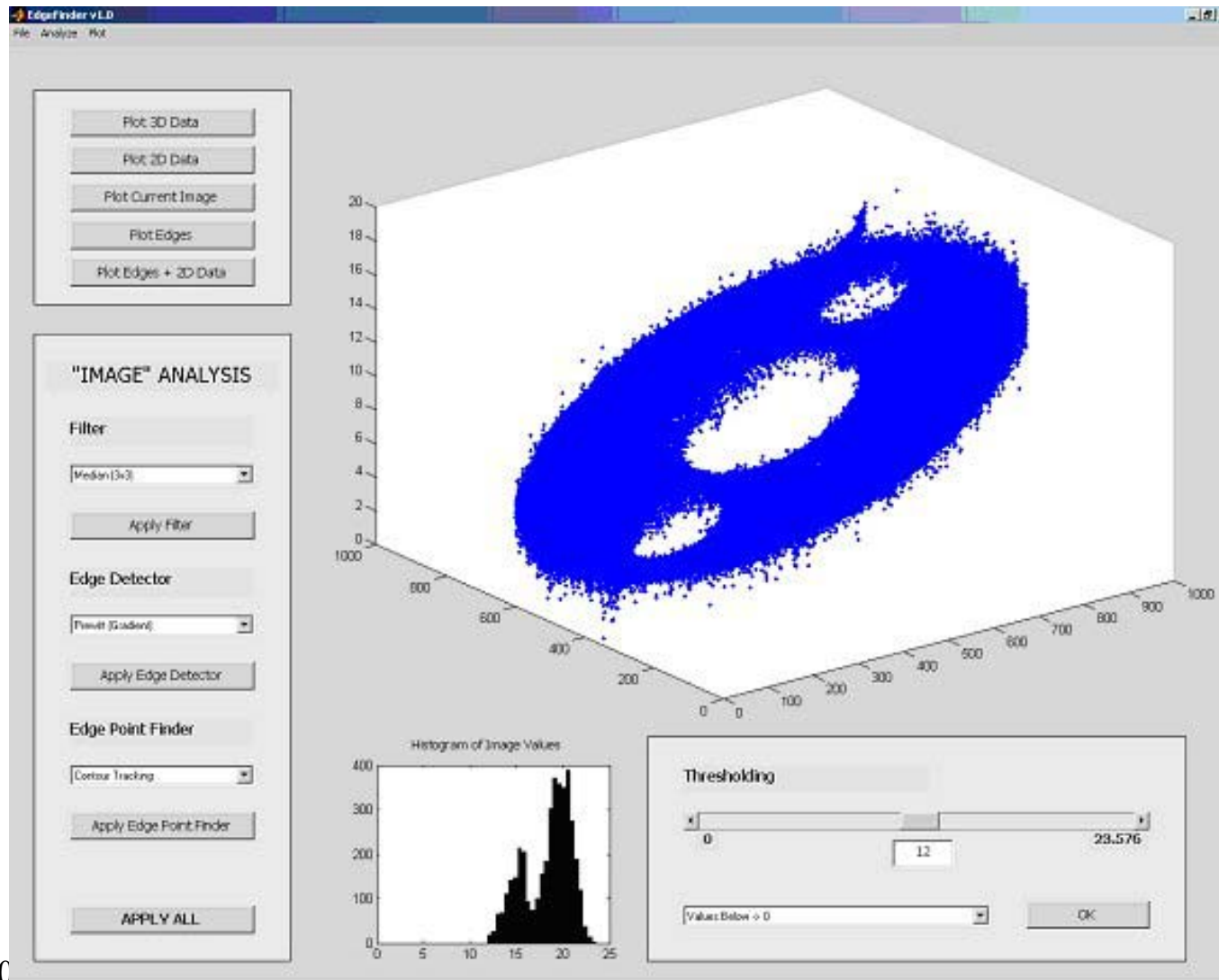
- ❖ Fit two dimensional data to lines, circles and ellipses
- ❖ If data was taken from the surface of an object, edges need to be found



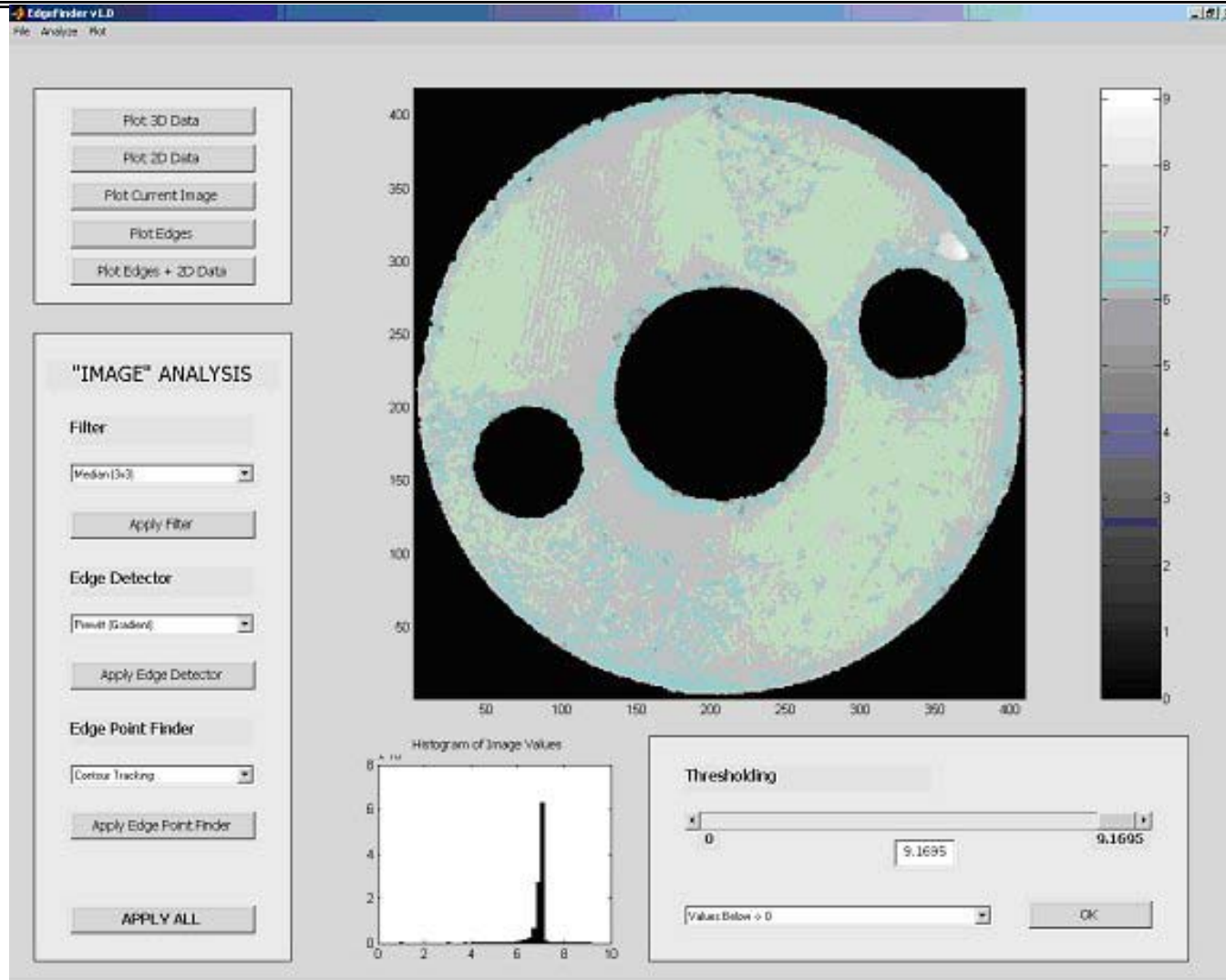
# Edge Location



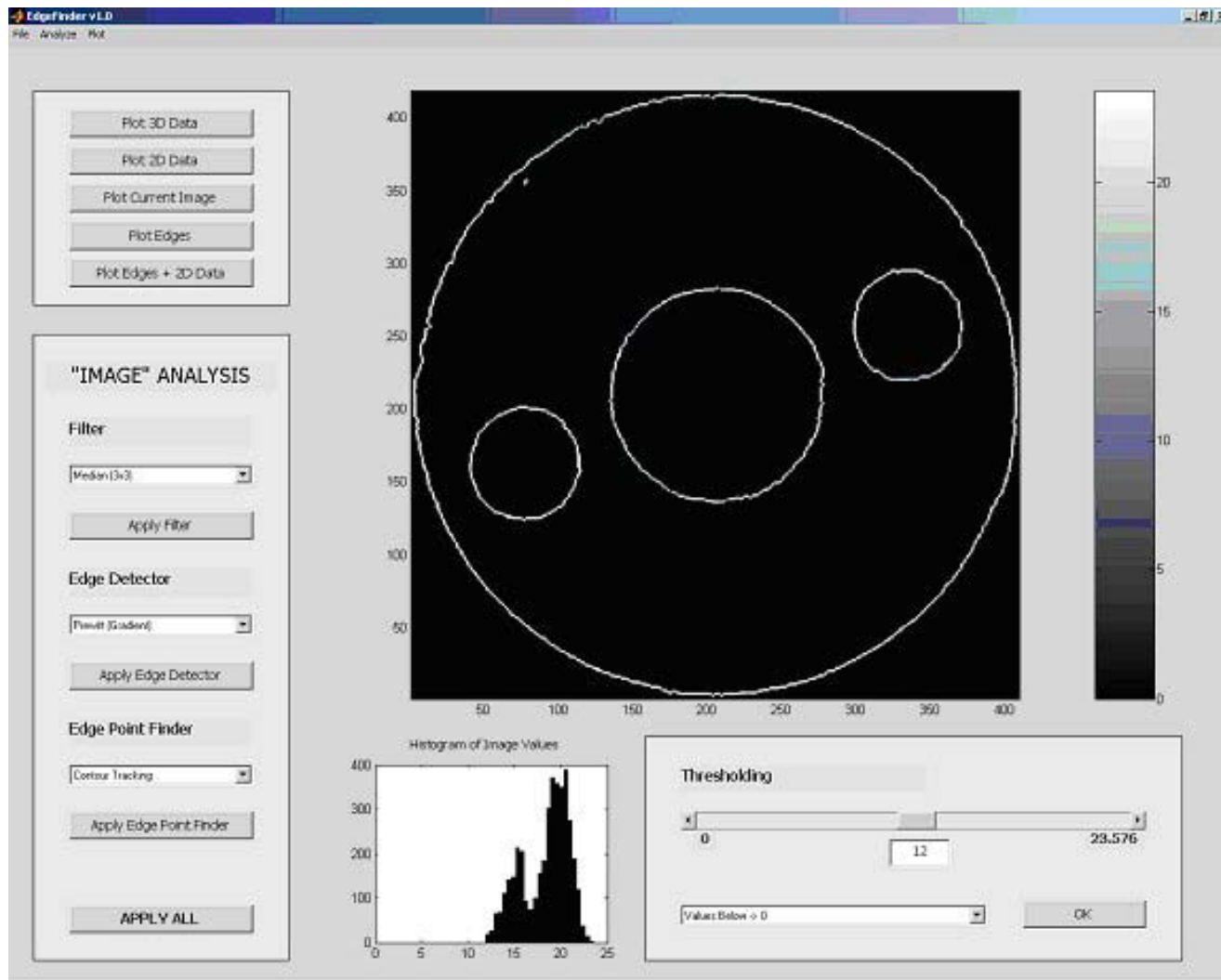
# Initial Point Cloud



# Filtered Image



# Edges from Edge Detector





# Resulting Edge Points

